Chapter 33 Performability Modeling of Distributed Systems and Its Formal Methods Representation

Razib Hayat Khan

American International University-Bangladesh, Bangladesh, Dhaka

ABSTRACT

A distributed system is a complex system. Developing complex systems is a demanding task when attempting to achieve functional and non-functional properties such as synchronization, communication, fault tolerance. These properties impose immense complexities on the design, development, and implementation of the system that incur massive effort and cost. Therefore, it is vital to ensure that the system must satisfy the functional and non-functional properties. Once a distributed system is developed, it is very difficult and demanding to conduct any modification in its architecture. As a result, the quantitative analysis of a complex distributed system at the early stage of the development process is always an essential and intricate endeavor. To meet the above challenge, this chapter introduces an extensive framework for performability evaluation of a distributed system. The goal of the performability modeling framework is to consider the behavioral change of the system components due to failures. This reveals how such behavioral changes affect the system performance.

INTRODUCTION

Conducting performance evaluation of a distributed system separately from the dependability evaluation fails to assess the anticipated system performance in the presence of system components failure and recovery. System dynamics is affected by any state changes of the system components due to failure and recovery. This chapter introduces the concept of performability that considers the behavioral change of the system components due to failures and also reveals how this behavioral change affects the system performance. But to design a composite model for a distributed system, perfect modeling of the overall system behavior is essential and sometimes very unwieldy. A distributed system behavior is normally realized by the several objects that are physically disseminated. The overall system behavior

DOI: 10.4018/978-1-7998-5339-8.ch033

is maintained by the partial behavior of the distributed objects of the system (Khan & Heegaard, 2011). So, it is essential to model the distributed objects behavior perfectly for appropriate demonstration of the system dynamics and to conduct the performability evaluation (Khan & Heegaard, 2011). Hence, we adopt Unified Modeling Language (UML) collaboration, state machine, deployment, and activity-oriented approach as UML is the most commonly used specification language which models both the system requirements and qualitative behavior through an assortment of notations (OMG, 2017; Khan & Heegaard, 2011). Our way of utilizing UML collaboration and activity diagram to capture the system dynamics, provides the opportunity to reuse the software components. The specifications of collaboration are given as coherent, self-contained building blocks (Khan & Heegaard, 2011). Reusability of the software component is achieved by designing the collaborative building block, which is used as main specification unit in this work.

In order to guarantee the precise understanding and correctness of the model, the approach requires formal reasoning on the semantics of the language used and to maintain the consistency of the models' specification. Temporal logic is a suitable option in this regard. In particular, the properties of super position supported by compositional temporal logic of action (cTLA) make it possible to describe systems from different viewpoints and by individual processes that are superimposed (Herrmann & Krumm, 2000). In this work, we focus on the cTLA that allows us formalizing the collaborative service specifications given by UML activities, deployment diagram and state machine (STM) model. By expressing collaborations as cTLA processes, we can ensure that a composed service maintains the properties of the individual collaborations it is comprised of. The semantic definition of collaboration, activity, deployment, and STM model in the form of temporal logic is implemented as a transformation tool which produces TLA⁺ modules. These modules may then be used as input for the temporal model checker (TLC) for syntactic analysis (Slåtten, 2007).

Furthermore, UML models are annotated according to the UML profile for Modeling and Analysis of Real- Time and Embedded Systems (MARTE) (OMG 2019) and UML profile for Modeling Quality of Service and Fault Tolerance Characteristics (OMG, 2008) to include quantitative system parameters necessary for performability evaluation. Considering the annotated UML models, UML specification styles are applied to generate SRN models automatically following the model transformation rules. The execution of performance SRN is dependent on the execution of dependability SRN. A transition in the dependability SRN may induce a state change in the performance SRN. The stepwise process to generate the performability SRN model is covered in this chapter.

Over decades several performability modeling techniques have been considered such as Markov models, Stochastic Petri Net (SPN), and Stochastic Reward Net (SRN) (Trivedi, 2011). Among all of these, we will focus on the SRN as performability model generated by our framework due to its prominent and interesting properties such as priorities assignment in transitions, presence of guard functions for enabling transitions that can use entire state of the net rather than a particular state, marking dependent arc multiplicity that can change the structure of the net, marking dependent firing rates, and reward rates defined at the net level.

The modeling process is supported by a set of tools, including Arctis and SHARPE with the incremental model checking facility (Khan, 2014). Arctis is used for specifying the system functional behavior. The evaluation of the performability models generated by the framework is achieved using SHARPE.

The objective of this chapter is to describe our novel approach that allows the evaluation of the performance and dependability related behavior of a distributed systems in a combined and automated way. The chapter is organized as follows: Next section defines the literature study, subsequent sections introduce 22 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/performability-modeling-of-distributed-systemsand-its-formal-methods-representation/275309

Related Content

Performability Modeling of Distributed Systems and Its Formal Methods Representation

Razib Hayat Khan (2021). Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing (pp. 704-727). www.irma-international.org/chapter/performability-modeling-of-distributed-systems-and-its-formal-methods-representation/275309

Cloud Computing as a Catalyst in STEM Education

Vikas Kumarand Deepika Sharma (2021). Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing (pp. 470-485). www.irma-international.org/chapter/cloud-computing-as-a-catalyst-in-stem-education/275297

Strategic Values of Cloud Computing Transformation: A Multi-Case Study of 173 Adopters

Mohamed Makhloufand Oihab Allal-Chérif (2021). Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing (pp. 1668-1684). www.irma-international.org/chapter/strategic-values-of-cloud-computing-transformation/275359

Understanding Intentions to Switch Toward Cloud Computing at Firms' Level: A Multiple Case Study in Tunisia

Zouhour Smaoui Hachichaand Karim Mezghani (2021). *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing (pp. 2415-2447).* www.irma-international.org/chapter/understanding-intentions-to-switch-toward-cloud-computing-at-firms-level/275397

Realm Towards Service Optimization in Fog Computing

Ashish Tiwariand Rajeev Mohan Sharma (2021). *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing (pp. 1530-1563).* www.irma-international.org/chapter/realm-towards-service-optimization-in-fog-computing/275353